JUSTIS Information System for the District of Columbia

Phase 2 Project file

JUSTIS Performance Report

Document History	
Document Filename:	Performance Report version 1.0
Document Type:	JUSTIS Phase 2 Performance Document
Document Status:	Prepared for ITLO Review
Time of Last Update:	5/4/2004 7:54 AM
Project Name:	DC JUSTIS Phase 2
Contract Number:	DC-C-920-S-065
Document Purpose:	This document details the performance of the JUSTIS Inquiry application
Revision History	5/09/2002 _ Draft Version: Pending Management Review 5/13/2002 – Reviewed by JUSTIS Engagement Manager

KPMG Consulting

Table of Contents

1. INTRODUCTION 2

1.1 PURPOSE OF THIS REPORT 2
1.2 AUDIENCE 2
1.3 PERFORMANCE IMPROVEMENT 2

2. JUSTIS INQUIRY APPLICATION ARCHITECTURE 3

2.1 INDIVIDUAL USER COMPUTERS 5
2.2 AGENCY FIREWALLS 5
2.3 DC WAN 9
2.3.1 PING TEST RESULTS FROM JUSTISHUB TO ALL AGENCY SERVERS: 9
2.3.2 PING TEST RESULTS FROM JUSTISCSOSA TO JUSTISHUB: 12
2.3.3 PING TEST RESULTS FROM JUSTISDCSC TO JUSTISHUB: 13
2.3.4 PING TEST RESULTS FROM JUSTISUSAO TO JUSTISHUB: 13
2.3.5 PING TEST RESULTS FROM JUSTISMPD TO JUSTISHUB: 14
2.3.6 PING TEST RESULTS FROM JUSTISUSPC TO JUSTISHUB: 15
2.3.7 PING TEST RESULTS FROM ITLO WORKSTATION TO JUSTISHUB: 16
2.4 CRYPTEK ENCRYPTION 17
2.5 SSL RENDITION 18
2.6 JUSTISHUB IIS WEB SERVER 18
2.7 JUSTISHUB DATABASE SERVER 19
2.8 JUSTIS INQUIRY APPLICATION 24

3. SUMMARY OF ANALYSIS 26

1. Introduction

1.1 Purpose of This Report

This report addresses the issues related to the performance of the JUSTIS Inquiry Application. It was brought to the attention of the JUSTIS team by the Information Technology Liaison Officer (ITLO) that the inquiry application sometimes takes an inordinate amount of time (between 45 seconds to 60 seconds) to respond to generic user queries. At the ITLO's request the JUSTIS team evaluated the application's performance by conducting similar sets of inquiries at four separate locations. The outcomes of these inquiries are documented in this report.

1.2 Audience

The intended audience for this report is JUSTIS management and systems support personnel and OCTO DC LAN/WAN administrators. The nature of the report is necessarily somewhat technical. A technical background, while useful, is not a pre-requisite to understanding the report or its findings.

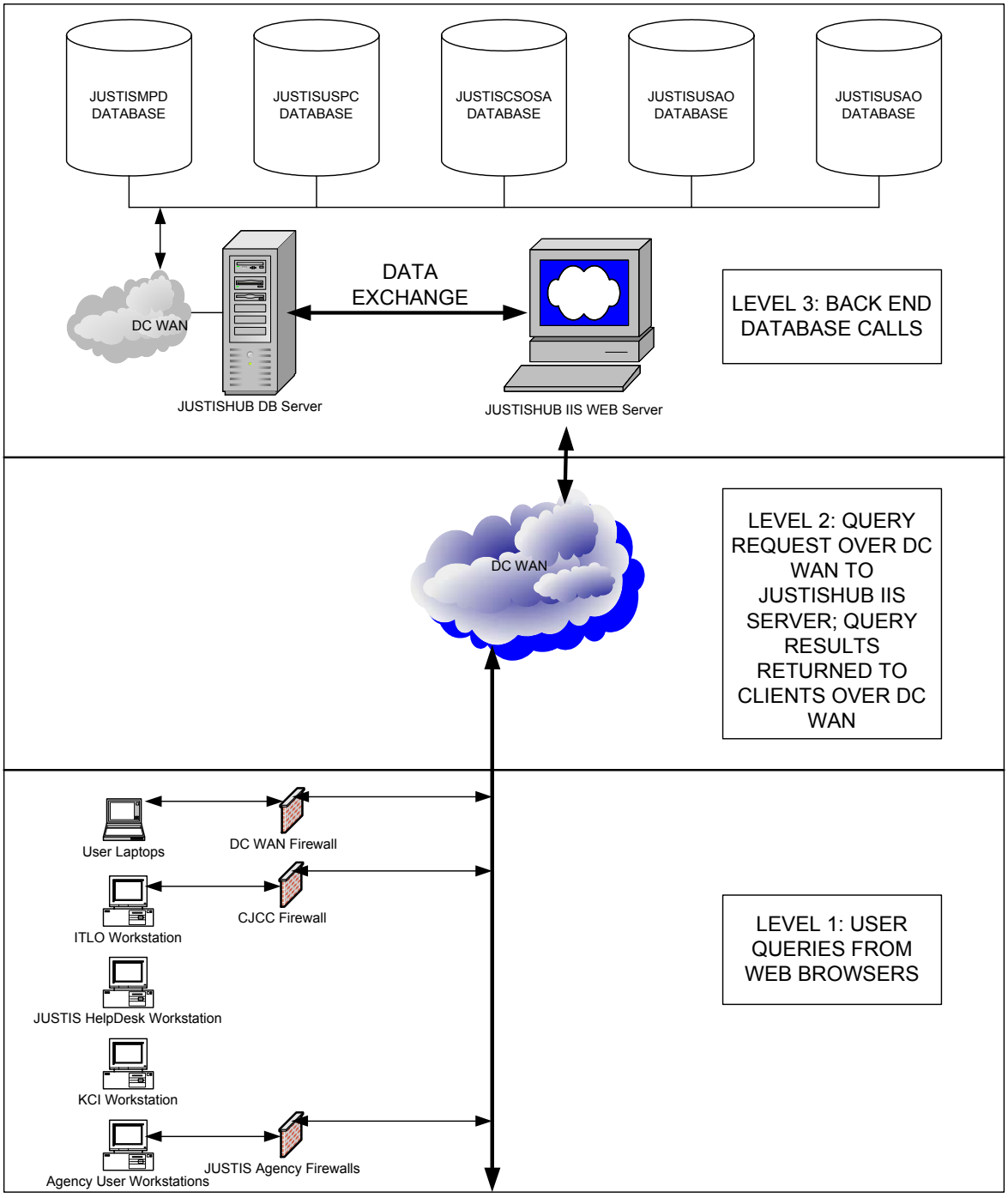
1.3 Performance Improvement

The operations and technical support staff of JUSTIS monitor performance in a number of ways. Through a process of on going monitoring, JUSTIS support staff can note trends and plan accordingly. Action can be taken to address performance degradation before it becomes a major issue for users.

Performance can degrade for a variety of reasons. Most notable among these are hardware failures (addressed under the JUSTIS Phase 2 document *"Hardware Expansion Plan"*), an increase in the number of users, and an increase in usage of the system.

The JUSTIS operations staff monitors performance in two ways. First, the JUSTIS inquiry application logs user actions, errors, and performance data. The operations staff monitors the performance table in JUSTIS, aggregating response times over a period of time and looking for indications of performance degradation.

2. JUSTIS Inquiry Application Architecture



The diagram above represents the current state of the JUSTIS Inquiry Application. Any user query against the Inquiry Application goes through an agency firewall to the DC WAN and submits its query request to the JUSTISHUB IIS Server over a secure socket layer.

The IIS Server queries the JUSTISHUB Database Server, which in turn queries various distributed agency databases over the DC WAN. The results of the JUSTISHUB database server query are returned to the IIS Server. The results of the query are returned to the user over an SSL channel.

It is apparent that there are a number of areas that could be potential performance degradation points (i.e. bottlenecks) in this process:

1. **Individual user computers**: User computers might have insufficient memory or other resources to connect to and effectively interrogate the JUSTISHUB IIS Server.
2. **Agency Firewalls**: Where agency firewalls exist (as they do at MPD, CSOSA and DCSC for example), users of that agency will experience additional delays, as user queries must be routed through the firewall out to the DC WAN and back through the firewall into the agency, instead of a possibly more direct route.
3. **DC WAN**: The DC WAN is a major component of the entire process. All user traffic is carried on this network and the volume of traffic at any given point in time could impact the speed with which a user query is executed. The performance of the DC WAN is dependent on existing hardware infrastructure beyond the control of the JUSTIS team. If certain network segments operate sub-optimally, JUSTIS Inquiries on those segments will be impacted.
4. **CRYPTEK Encryption**: The sensitive nature of Criminal Justice data necessitates the encryption of queries and data in transit between the JUSTISHUB Server and the Agency database servers. The CRYPTEK solution is employed for the purpose and if not operating correctly, could impact query response times.
5. **SSL rendition**: The sensitive nature of Criminal Justice data necessitates SSL encryption of all user queries, their decryption at the JUSTISHUB for processing, re-encryption of the query results for transmission to users and another decryption at the user computer. This process adds additional overhead to query times.
6. **JUSTISHUB Web Server**: The JUSTISHUB IIS Server is a key component of the Inquiry process. If this server were to be offline or otherwise operating below peak efficiency, query response times would be impacted.
7. **JUSTISHUB Database Server**: The JUSTISHUB database server should be tuned to effectively query the distributed agency databases and return results to the IIS Server correctly and efficiently. If this is not performed correctly, query response times could be impacted.
8. **JUSTIS Inquiry Application**: Unless the JUSTIS Inquiry Application is tuned to operate at peak efficiency, query response times could be impacted.

The remaining sections of this document seek to present clear evidence (or the lack thereof) that would help to pinpoint critical areas that could lead to query performance improvements. A breakup by category of typical query response times is also presented in the Findings Section.

2.1 Individual User Computers

In general, individual user computers offer a clear indication of performance problems or the lack thereof.

User computers might be limited by memory, processing capability or available disk space when running queries.

However, none of these are deemed to be limiting factors for the JUSTIS User community.

The current hardware market has rendered anything Wintel platform below a Pentium II essentially obsolete. A cursory survey of the computers employed within the DC Government indicates that most computers meet and exceed these standards. Most user computers have 256Mb+ of RAM, a Pentium III or higher processor at 300Mhz and at least 300Mb of available disk space. Regular virus checks and disk defragmentation runs will ensure that a given user computer is capable of launching a web browser, connecting to a Web Server and retrieving web pages from this Web Server.

Provided the user computer meets these minimum standards, any performance issues must be attributed to the Web Server, network or other system component.

A simple test for individual user computers is to test the computer against other web sites. If query data from any other web site (such as a word search query run on Google for example) returns much faster than a query on the Inquiry Application, then the user computer is not the issue.

The JUSTIS team did not find any user computer that was seriously below standards.

Where the user computer is the problem, any performance overhead attributable to this factor is estimated to be at most 2 additional seconds.

2.2 Agency Firewalls

Agency firewalls and proxy web servers exist at a number of the constituent JUSTIS user agencies including MPD, CSOSA and DCSC. These have been set up by the agencies to manage, monitor, protect or otherwise route network traffic in a specific and known manner.

The JUSTIS Inquiry application has to work within the confines of this pre-existing agency infrastructure. User queries from these agencies are routed through additional computers or network segments (additional network hops) and query results must travel through these additional segments to be returned to the users.

This process causes response time overheads for users at these agencies. Where computers/routers on these specific segments are down, user queries might take longer to be executed over alternative routes or may fail altogether. These are factors controlled by the individual agencies. A change in the Inquiry Application will not alleviate this situation.

The JUSTIS team performed tests to trace the route taken by network traffic relevant to JUSTIS Inquiries. The data from these tests is reproduced below:

Network traces were done from the OCTO Data Center to the agency servers and vice versa. The tables below summarize the network path and hops it took to reach the destination server. The time taken for the hops is recorded in milliseconds[1].

From JUSTISHUB to JUSTISCSOSA	
JUSTISHUB	0
164.82.72.15	5
164.82.10.34	8
164.82.34.2	7
208.245.205.12	7
10.0.81.1	8
JUSTISCSOSA	8
Total	43 milliseconds

From JUSTISCSOSA TO JUSTISHUB	
JUSTISTICSOSA	0
207.16.184.1	7
10.0.81.2	5
208.245.205.8	5
164.82.34.1	5
164.82.10.40	5
JUSTISHUB	8
Total	35 milliseconds

[1] Note: Actual times were double those presented here. Actual times were Total Loop (TTL) times, i.e. the amount of time it takes a packet to go *from* the pinging server *to* the pinged server and *return* to the pinging server. Since a trace analysis only requires a one-way analysis of the route, times have been halved for presentation here.

FROM JUSTISHUB TO JUSTISMPD	
JUSTISHUB	0
164.82.72.15	5
164.82.72.34	8
164.82.10.30	8
164.82.194.33	15
JUSTISMPD	16
Total	52 milliseconds

FROM JUSTISMPD TO JUSTISHUB	
JUSTISMPD	0
164.82.194.97	8
164.82.194.35	6
164.82.10.1	15
164.82.220.3	8
164.82.10.40	8
JUSTISHUB	15
Total	60 milliseconds

FROM JUSTISHUB TO JUSTISUSPC	
JUSTISHUB	0
164.82.72.15	8
164.82.10.45	8
10.14.46.1	15
10.103.0.4	8
JUSTISUSPC	8
Total	47 milliseconds

FROM JUSTISUSPC TO JUSTISHUB	
JUSTISUSPC	0
10.103.0.4	7
10.14.46.1	8
164.82.10.40	15
JUSTISHUB	8
Total	38 milliseconds

The tests above were done to measure the effect, if any, of the network on the JUSTIS Inquiry Application's response time. The hypothesis was that the more convoluted the route to JUSTISHUB taken by a query over the network, the longer the response time would be. This is borne out by the data.

This methodology does not apply to the JUSTISDCSC and JUSTISUSAO servers. Both servers are located at the OCTO data center itself and connected to the same segment as the JUSTISHUB server. Consequently, when querying JUSTISHUB over the network, both servers returned response times below 10 milliseconds, with no network hops, consistent with a direct connection to JUSTISHUB.

LOCATIONAL TRACER: NETWORK TRAFFIC FROM 441 JUDICIARY SQ. TO JUSTISHUB	
10.128.12.2	5
gw-10.128.0.dc.gov (10.128.0.3)	5
164.82.10.40	6
JUSTISHUB	5
Total	21 milliseconds

LOCATIONAL TRACER: NETWORK TRAFFIC FROM ITLO'S WORKSTATION TO JUSTISHUB	
10.28.30.3	1
10.28.0.4	1
164.82.10.48	3
JUSTISHUB	4
Total	9 milliseconds

Notwithstanding this data, queries put to individual agencies by the JUSTIS team have revealed that most agencies do not employ multiple firewalls or proxy servers. Therefore, while the Inquiry Application must operate within the environment presented above, query response delays due to this factor are estimated to be at most 2 additional seconds.

2.3 DC WAN

User queries initiated at the Inquiry Application travel through the DC WAN to the JUSTISHUB Server. The JUSTISHUB Server processes these queries across multiple distributed agency databases through the DC WAN. The processed queries are then returned to the user computer through the DC WAN.

It is apparent, therefore, that any performance issues with the DC WAN will have a cascading effect on the Inquiry Application.

The JUSTIS team has evaluated network traffic from the JUSTISHUB Server to the agency servers, from user computers to the JUSTISHUB Server and from the agency servers to the JUSTISHUB Server. The use of a network ping utility, run automatically every two hours over a period of two to four weeks has confirmed that the performance of the DC WAN is generally consistent over time. Representative statistics from the ping tests are reproduced below:

2.3.1 PING TEST RESULTS FROM JUSTISHUB TO ALL AGENCY SERVERS:

Wed 04/24/2002 2:00p

Pinging **JUSTISCSOSA** [207.16.185.11] with 32 bytes of data:
Reply from 207.16.185.11: bytes=32 time=16ms TTL=123
Reply from 207.16.185.11: bytes=32 time<10ms TTL=123
Reply from 207.16.185.11: bytes=32 time=16ms TTL=123

Reply from 207.16.185.11: bytes=32 time=16ms TTL=123
Ping statistics for 207.16.185.11:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 16ms, **Average = 12ms**

Pinging **JUSTISMPD** [164.82.194.107] with 32 bytes of data:
Reply from 164.82.194.107: bytes=32 time=15ms TTL=123
Reply from 164.82.194.107: bytes=32 time=31ms TTL=123
Reply from 164.82.194.107: bytes=32 time=31ms TTL=123
Reply from 164.82.194.107: bytes=32 time=16ms TTL=123
Ping statistics for 164.82.194.107:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 15ms, Maximum = 31ms, **Average = 23ms**

Pinging **JUSTISUSAO** [164.82.72.93] with 32 bytes of data:
Reply from 164.82.72.93: bytes=32 time<10ms TTL=128
Reply from 164.82.72.93: bytes=32 time<10ms TTL=128
Reply from 164.82.72.93: bytes=32 time<10ms TTL=128
Reply from 164.82.72.93: bytes=32 time<10ms TTL=128
Ping statistics for 164.82.72.93:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, **Average = 0ms**

Pinging **JUSTISDCSC** [164.82.72.94] with 32 bytes of data:
Reply from 164.82.72.94: bytes=32 time<10ms TTL=128
Reply from 164.82.72.94: bytes=32 time=16ms TTL=128
Reply from 164.82.72.94: bytes=32 time<10ms TTL=128
Reply from 164.82.72.94: bytes=32 time<10ms TTL=128
Ping statistics for 164.82.72.94:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 16ms, **Average = 4ms**

Pinging **JUSTISUSPC** [10.14.46.9] with 32 bytes of data:
Reply from 10.14.46.9: bytes=32 time=16ms TTL=125

Reply from 10.14.46.9: bytes=32 time=16ms TTL=125
Reply from 10.14.46.9: bytes=32 time=16ms TTL=125
Reply from 10.14.46.9: bytes=32 time=16ms TTL=125
Ping statistics for 10.14.46.9:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:

Minimum = 16ms, Maximum = 16ms, **Average = 16ms**

Wed 04/24/2002 4:00p

Pinging **JUSTISCSOSA** [207.16.185.11] with 32 bytes of data:
Reply from 207.16.185.11: bytes=32 time=15ms TTL=123

Reply from 207.16.185.11: bytes=32 time=15ms TTL=123
Reply from 207.16.185.11: bytes=32 time=15ms TTL=123
Reply from 207.16.185.11: bytes=32 time=15ms TTL=123
Ping statistics for 207.16.185.11:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 15ms, Maximum = 15ms, **Average = 15ms**

Pinging **JUSTISMPD** [164.82.194.107] with 32 bytes of data:
Reply from 164.82.194.107: bytes=32 time=15ms TTL=123
Reply from 164.82.194.107: bytes=32 time=15ms TTL=123
Reply from 164.82.194.107: bytes=32 time=16ms TTL=123
Reply from 164.82.194.107: bytes=32 time=16ms TTL=123
Ping statistics for 164.82.194.107:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:
Minimum = 15ms, Maximum = 16ms, **Average = 15ms**

Pinging **JUSTISUSAO** [164.82.72.93] with 32 bytes of data:
Reply from 164.82.72.93: bytes=32 time<10ms TTL=128
Reply from 164.82.72.93: bytes=32 time<10ms TTL=128
Reply from 164.82.72.93: bytes=32 time<10ms TTL=128
Reply from 164.82.72.93: bytes=32 time<10ms TTL=128
Ping statistics for 164.82.72.93:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, **Average = 0ms**

Pinging **JUSTISDCSC** [164.82.72.94] with 32 bytes of data:
Reply from 164.82.72.94: bytes=32 time<10ms TTL=128
Reply from 164.82.72.94: bytes=32 time<10ms TTL=128
Reply from 164.82.72.94: bytes=32 time<10ms TTL=128
Reply from 164.82.72.94: bytes=32 time<10ms TTL=128
Ping statistics for 164.82.72.94:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, **Average = 0ms**

Pinging **JUSTISUSPC** [10.14.46.9] with 32 bytes of data:
Reply from 10.14.46.9: bytes=32 time=16ms TTL=125
Reply from 10.14.46.9: bytes=32 time=32ms TTL=125
Reply from 10.14.46.9: bytes=32 time=16ms TTL=125

Reply from 10.14.46.9: bytes=32 time=32ms TTL=125
Ping statistics for 10.14.46.9:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 16ms, Maximum = 32ms, **Average = 24ms**

It is apparent from these statistics that no significant variation exists in the network traffic between JUSTISHUB and the agency servers regardless of time or day.

2.3.2 PING TEST RESULTS FROM JUSTISCSOSA TO JUSTISHUB:

Fri 04/19/2002 10:51a

Pinging **JUSTISHUB** [164.82.72.80] with 32 bytes of data:
Reply from 164.82.72.80: bytes=32 time=16ms TTL=123
Reply from 164.82.72.80: bytes=32 time<10ms TTL=123
Reply from 164.82.72.80: bytes=32 time<10ms TTL=123
Reply from 164.82.72.80: bytes=32 time<10ms TTL=123
Ping statistics for 164.82.72.80:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 16ms, **Average = 4ms**

Wed 05/08/2002 12:47p

Pinging **JUSTISHUB** [164.82.72.80] with 32 bytes of data:
Reply from 164.82.72.80: bytes=32 time=16ms TTL=123
Reply from 164.82.72.80: bytes=32 time<10ms TTL=123
Reply from 164.82.72.80: bytes=32 time<10ms TTL=123
Reply from 164.82.72.80: bytes=32 time<10ms TTL=123
Ping statistics for 164.82.72.80:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 16ms, **Average = 4ms**

It is apparent from these statistics that no significant variation exists in the network traffic between JUSTISCSOSA and the JUSTISHUB regardless of time or day.

2.3.3 PING TEST RESULTS FROM JUSTISDCSC TO JUSTISHUB:

Fri 04/19/2002 10:00a

Pinging **JUSTISHUB.justis.dc.gov** [164.82.72.80] with 32 bytes of data:
Reply from 164.82.72.80: bytes=32 time=16ms TTL=128

Reply from 164.82.72.80: bytes=32 time<10ms TTL=128
Reply from 164.82.72.80: bytes=32 time<10ms TTL=128
Reply from 164.82.72.80: bytes=32 time<10ms TTL=128
Ping statistics for 164.82.72.80:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 16ms, **Average = 4ms**

Sat 04/20/2002 8:00a

Pinging **justishub.justis.dc.gov** [164.82.72.80] with 32 bytes of data:
Reply from 164.82.72.80: bytes=32 time<10ms TTL=128
Reply from 164.82.72.80: bytes=32 time<10ms TTL=128
Reply from 164.82.72.80: bytes=32 time<10ms TTL=128
Reply from 164.82.72.80: bytes=32 time<10ms TTL=128
Ping statistics for 164.82.72.80:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, **Average = 0ms**

It is apparent from these statistics that no significant variation exists in the network traffic between JUSTISDCSC and the JUSTISHUB regardless of time or day.

2.3.4 PING TEST RESULTS FROM JUSTISUSAO TO JUSTISHUB:

Fri 04/19/2002 10:00a

Pinging **JUSTISHUB** [164.82.72.80] with 32 bytes of data:
Reply from 164.82.72.80: bytes=32 time=16ms TTL=128
Reply from 164.82.72.80: bytes=32 time<10ms TTL=128
Reply from 164.82.72.80: bytes=32 time<10ms TTL=128
Reply from 164.82.72.80: bytes=32 time<10ms TTL=128

Ping statistics for 164.82.72.80:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 16ms, **Average = 4ms**

Sat 04/20/2002 8:00a

Pinging **JUSTISHUB** [164.82.72.80] with 32 bytes of data:
Reply from 164.82.72.80: bytes=32 time<10ms TTL=128
Reply from 164.82.72.80: bytes=32 time<10ms TTL=128
Reply from 164.82.72.80: bytes=32 time<10ms TTL=128
Reply from 164.82.72.80: bytes=32 time<10ms TTL=128
Ping statistics for 164.82.72.80:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, **Average = 0ms**

It is apparent from these statistics that no significant variation exists in the network traffic between JUSTISUSAO and the JUSTISHUB regardless of time or day.

2.3.5 PING TEST RESULTS FROM JUSTISMPD TO JUSTISHUB:

Fri 04/19/2002 3:05p

Pinging **JUSTISHUB** [164.82.72.80] with 32 bytes of data:

Reply from 164.82.72.80: bytes=32 time=32ms TTL=124

Reply from 164.82.72.80: bytes=32 time=16ms TTL=124

Reply from 164.82.72.80: bytes=32 time=16ms TTL=124

Reply from 164.82.72.80: bytes=32 time=16ms TTL=124

Ping statistics for 164.82.72.80:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 16ms, Maximum = 32ms, **Average = 20ms**

Wed 05/08/2002 3:45p

Pinging **JUSTISHUB** [164.82.72.80] with 32 bytes of data:

Reply from 164.82.72.80: bytes=32 time=15ms TTL=124

Reply from 164.82.72.80: bytes=32 time=15ms TTL=124

Reply from 164.82.72.80: bytes=32 time=15ms TTL=124

Reply from 164.82.72.80: bytes=32 time=16ms TTL=124

Ping statistics for 164.82.72.80:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 15ms, Maximum = 16ms, **Average = 15ms**

It is apparent from these statistics that no significant variation exists in the network traffic between JUSTISMPD and the JUSTISHUB regardless of time or day.

2.3.6 PING TEST RESULTS FROM JUSTISUSPC TO JUSTISHUB:

Mon 04/29/2002 2:49p

Pinging **JUSTISHUB** [164.82.72.80] with 32 bytes of data:

Destination host unreachable.

Destination host unreachable.

Destination host unreachable.

Destination host unreachable

Ping statistics for 164.82.72.80:

Packets: Sent = 4, Received = 0, Lost = 4 **(100% loss),**

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 0ms, **Average = 0ms**

Mon 04/29/2002 4:00p

Pinging **JUSTISHUB** [164.82.72.80] with 32 bytes of data:

Request timed out.

Request timed out.

Reply from 164.82.72.80: bytes=32 time=10ms TTL=124

Reply from 164.82.72.80: bytes=32 time=10ms TTL=124

Ping statistics for 164.82.72.80:

Packets: Sent = 4, Received = 2, Lost = 2 **(50% loss)**,

Approximate round trip times in milli-seconds:

Minimum = 10ms, Maximum = 10ms, **Average = 5ms**

Mon 04/29/2002 6:00p

Pinging **JUSTISHUB** [164.82.72.80] with 32 bytes of data:

Request timed out.

Request timed out.

Request timed out.

Reply from 164.82.72.80: bytes=32 time=10ms TTL=124

Ping statistics for 164.82.72.80:

Packets: Sent = 4, Received = 1, Lost = 3 **(75% loss)**,

Approximate round trip times in milli-seconds:

Minimum = 10ms, Maximum = 10ms, **Average = 2ms**

While it is apparent from these statistics that no significant variation exists in the network traffic between JUSTISUSPC and the JUSTISHUB regardless of time or day, it is also clear from the packet loss that some connectivity problems exist between USPC and the JUSTISHUB. This is reflected in the increased time taken for USPC query responses (see Section 2.7), but is not considered a seriously limiting factor.

2.3.7 PING TEST RESULTS FROM ITLO WORKSTATION TO JUSTISHUB:

Thurs 04/25/2002 9:17a

Pinging **JUSTISHUB** [164.82.72.80] with 32 bytes of data:
Reply from 164.82.72.80: bytes=32 time=7ms TTL=125

Reply from 164.82.72.80: bytes=32 time=6ms TTL=125
Reply from 164.82.72.80: bytes=32 time=6ms TTL=125
Reply from 164.82.72.80: bytes=32 time=4ms TTL=125
Ping statistics for 164.82.72.80:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 4ms, Maximum = 7ms, **Average = 5ms**

It is apparent from these statistics that network performance from the ITLO Workstation is in line with the network performance that exists between the JUSTISHUB and other agency servers/users.

This analysis can be extended further to estimate JUSTIS data transfer times over the DC WAN.

A typical JUSTIS Inquiry (such as a query for a single agency where PDID is like '123%') from a user computer to the JUSTISHUB Server results in the transfer of a 10Kb file over the network.

A query being processed between the JUSTISHUB server and an agency database server results in the transfer of about 1Kb of data over the network.

A representative 32b ping packet takes about 8ms on average to travel back and forth between the JUSTISHUB Server and a single agency server or between the JUSTISHUB Server and a user computer.

Extrapolation indicates that data transfer between the JUSTISHUB server and agency database servers takes about .25 seconds. Data transfer between the JUSTISHUB Server and a user computer takes about 2.5 seconds. Therefore, the network data transfer time component of a query run against agencies is about 2.75 seconds per agency, per query.

2.4 CRYPTEK Encryption

The sensitive nature of Criminal Justice data necessitates encryption of JUSTIS queries and data in transit between the JUSTISHUB server and Agency database servers. This process adds additional overhead to query times.

The JUSTIS Inquiry Application employs the proprietary "CRYPTEK" solution developed by CRYPTEK Inc. to encrypt

In general, traffic through the CRYPTEK coding/decoding server does not add any significant response time overhead to the Inquiry process. The encryption/decryption process is estimated to add at most 0.5 seconds to query processing.

However, the JUSTIS team had noticed that the CRYPTEK solution suffers from some serious performance drawbacks.

When the CRYPTEK solution is operating normally, a typical inquiry (PDID like '123%') run across all JUSTIS agencies returns results in a total of about 18 seconds.

However, the CRYPTEK server sometimes inexplicably freezes locking JUSTIS agency servers out of the network. This action causes query response times to rise from the expected 18 seconds for a typical query to 45+ seconds.

The JUSTIS team has run tests where response times for a typical query have exceeded 1 minute under these circumstances.

The symptoms of this problem are well recognized within the JUSTIS team. Response times suddenly become unreasonably high and the JUSTIS WatchDog application indicates that the agency database server in question has lost network connectivity (when in fact, the server is physically very much on the network, but has been “frozen” out of it by CRYPTEK).

The short-term solution adopted by the JUSTIS team is to reboot each of the nodes on the CRYPTEK server. This frees up the agency servers and gets query response times back to normal.

The JUSTIS team also recommends a proactive weekly reboot of the CRYPTEK server nodes in an effort to stave off problems before they occur.

The long-term solution to this problem involves working with CRYPTEK Inc. to identify and correct the cause of this problem.

Should CRYPTEK Inc. be unable to satisfactorily explain and resolve this problem, the JUSTIS team would recommend the adoption of an alternative security solution.

2.5 SSL Rendition

The sensitive nature of Criminal Justice data necessitates SSL encryption of all user queries, their decryption at the JUSTISHUB for processing, re-encryption of the query results for transmission to users and another decryption at the user computer. This process adds additional overhead to query times. In general, this process is not expected to add more than a second to query response times.

2.6 JUSTISHUB IIS Web Server

All user query requests are made to the JUSTISHUB IIS Server. This server translates user requests and sends them to the JUSTISHUB database server for processing. On receiving the results of the query, the IIS Server transmits these back to the requesting user.

Should this server become overwhelmed by user traffic or otherwise operate below peak efficiency, response times could be compromised.

Tests performed by the JUSTIS team show that this server is performing at peak efficiency. The server has been configured to service over 100,000 user requests per day and 1000 user connections. The current JUSTIS User community is far below these capacity figures and the near future poses no problems.

Additional tests performed by the JUSTIS team in unloading the Inquiry Application and rebooting the IIS Server during a typical user session have yielded no significant change in query response performance figures.

Service request response times from the IIS Server to JUSTIS users are in the 0.5 to 1 second range for typical queries per agency.

In order to prepare for the future, the JUSTIS team recommends that an upgrade to IIS version 6.0 or a change of Web Server to Apache be performed when the user population doubles from its current levels.

Additionally, the JUSTISHUB Server currently doubles as a database server and web server. In order to maximize performance, a single machine dedicated to the Web Server function would be in order.

2.7 JUSTISHUB Database Server

Once the IIS Server passes a query on to the JUSTISHUB database server, the question of database efficiency is imperative. The JUSTISHUB database server and distributed agency servers should be tuned to return results to the IIS Server correctly and efficiently. If this is not performed correctly, query response times could be impacted.

Databases are tuned by managing indexes on tables to ensure that searches are performed efficiently and quickly. However, all other things being equal, a table with less data in it will return results faster than a table with a larger dataset even if both are indexed to the same level.

All queryable agency tables in the JUSTIS Inquiry application are completely indexed. The tables below summarize the indexes that are currently in use:

SQL Server: JUSTISCSOSA Database: NEWCSOSA Table: PAROLE

Index	Column
PK_PAROLE_KEY	PKEY
IX_PAROLE_DCDC	DCDC
IX_PAROLE_PDID	PDID
IX_PAROLE_SSN	SSN
IX_PAROLE_LN	LN
IX_PAROLE_FN	FN
IX_CASE_NO	CASE_NO

SQL Server: JUSTISCSOSA Database: NEWPSA Table: PRETRIAL

Index	Column
PK_PRETRIAL	PKEY
IX_PRETRIAL_PDID	PDID
IX_PRETRIAL_LN	LN
IX_PRETRIAL_FN	FN

SQL Server: JUSTISDCSC Database: YSA Table: YSA

Index	Column
PK_YSA	PKEY
IX_YSA_LN	LN
IX_YSA_FN	FN
IX_YSA_SF	SOCIAL FILE
IX_YSA_SSN	SSN

SQL Server: JUSTISDCSC Database: DCSC Table: CASEDETAIL

Index	Column
PK_CASEDETAIL	PKEY
IX_CASEDETAIL_CASE_NO	CASE_NO
IX_CASEDETAIL_LN	LN
IX_CASEDETAIL_FN	FN
IX_CASEDETAIL_PDID	PDID
IX_CASEDETAIL_ARN	ARN

SQL Server: JUSTISHUB Database: OCC Table: CASEDETAIL

Index	Column
PK_CASEDETAIL	PKEY
IX_OCC_CASE_NO	CASE_NO

SQL Server: JUSTISMPD Database: MPD Table: IDENTIFICATION

Index	Column
PK_IDENTIFICATION	PKEY
IX_IDENTIFICATION_PDID	PDID
IX_IDENTIFICATION_LN	LN
IX_IDENTIFICATION_FN	FN
IX_IDENTIFICATION_SSN	SSN
IX_IDENTIFICATION_CCN	CCN
IX_IDENTIFICATION_FBI	FBI
IX_IDENTIFICATION_PDID	PDID

SQL Server: JUSTISMPD Database: MPD Table: ARREST

Index	Column
PK_IDENTIFICATION_PDID	PKEY
IX_IDENTIFICATION_ARN	ARN
IX_IDENTIFICATION_CCN	CCN

SQL Server: JUSTISHUB Database: PDS Table: PDS

Index	Column
PK_PDS	PKEY
IX_PDS_BAR_NO	BAR_NO

SQL Server: JUSTISUSAO Database: USAO Table: CASEASSIGNMENT

Index	Column
PK_CASEASSIGNMENT	PKEY
IX_CASEASSIGNMENT_CASE_NO	CASE_NO
IX_CASEASSIGNMENT_LN	LN
IX_CASEASSIGNMENT_FN	FN
IX_IDENTIFICATION_PDID	PDID

SQL Server: JUSTISUSPC Database: USPC Table: DOCUMENT

Index	Column
PK_DOCUMENT_REG	REG_NO

SQL Server: JUSTISUSPC Database: USPC Table: PRISONER

Index	Column
PK_REG	REG
IX_PDID	PDID
IX_DCDC	DCDC
IX_LN	LN
IX_FN	FN

The proof of efficient indexing across agency databases is seen in the embedded HTML code extracted from the Inquiry Application.

A number of intentionally generic queries were run on the data from the constituent JUSTIS agencies. In each case, a count of the number of rows conforming to the search criteria was retrieved quickly. The embedded code, the row count returned and the agency database in question are seen below:

<!--amohan 5/8/2002 2:04:47 PM Ident Count took **308 milliseconds** for: Select COUNT(PDID) From Identification Where PDID Like '1%' --> **21015 rows in MPD Identification table**

<!--amohan 5/8/2002 2:04:50 PM Arrest Count took **125 milliseconds** for: Select COUNT(PDID) From Arrest Where PDID Like '1%' --> **508 rows in MPD Arrest table**

<!--amohan 5/8/2002 2:04:52 PM Par/Prb Count took **105 milliseconds** for: Select COUNT(PDID) From Parole Where PDID Like '1%' --> **2012 rows in CSOSA database**

<!--amohan 5/8/2002 2:04:54 PM Pretrial Count took **110 milliseconds** for: Select COUNT(PDID) From Pretrial Where PDID Like '1%' --> **11916 rows in PSA database**

<!--amohan 5/8/2002 2:05:00 PM Pris Count took **1231 milliseconds** for: Select COUNT(PDID) From Prisoner Where PDID Like '1%' --> **723 rows in USPC database**

<!--amohan 5/8/2002 2:08:58 PM Juv Count took **66 milliseconds** for: Select COUNT(SSN) From YSA Where SSN Like '1%' --> **2 rows in YSA database**

<!--amohan 5/8/2002 2:12:06 PM Bar # Count took **5 milliseconds** for: Select COUNT(BAR_NO) From PDS Where BAR_NO Like '2%' --> **2 rows in PDS database**

<!--amohan 5/8/2002 2:21:25 PM Case Count took **22 milliseconds** for: Select COUNT(CASE_NO) From CaseAssignment Where CASE_NO Like 'M%' --> **7206 rows in USAO database**

<!--amohan 5/8/2002 2:21:28 PM Case Count took **152 milliseconds** for: Select COUNT(CASE_NO) From CASEDETAIL Where CASE_NO Like 'M%' --> **89906 rows in DCSC database**

<!--amohan 5/8/2002 2:24:07 PM Case Count took **5 milliseconds** for: Select COUNT(CASE_NO) From CASEDETAIL Where CASE_NO Like 'D%' --> **2116 rows in OCC database**

The response times above range from 0.005 seconds (OCC and PDS) to 1.231 seconds (USPC), all extremely acceptable. Even the largest response time at USPC is attributable to the large volume of data in the USPC database, its physical distance from the JUSTISHUB database server and sometimes questionable network performance not the absence of good indexing and tuning. (See JUSTISUSPC ping statistics,2.3.6),

2.8 JUSTIS Inquiry Application

The JUSTIS Inquiry Application has been developed in Visual Studio 6.0 and uses Visual Basic 6.0 code and Webclasses. This application accepts user Inquiry requests, transmits them to the IIS Server, receives IIS responses and formats them as user query results for display within user Web browsers.

Given that the application is developed in Visual Studio 6.0, the JUSTIS Inquiry Application inherits the single-threaded operation functionality that applications written in this development environment are limited to.

From a practical stand point, what this means for a JUSTIS Inquiry user is that queries that run across multiple agencies are processed serially, not in parallel.

For example, if a user were to run a typical Inquiry, where PDID begins with 123, across all JUSTIS agencies, the Inquiry Application translates this request into six queries to be run across five agencies that allow searches by PDID (additionally, DC Department of Corrections allows searches by PDID, but the Inquiry application is not required to translate any request to DCDC since a real time direct connection exists).

These six queries are:

SELECT COUNT(PDID) FROM IDENTIFICATION WHERE PDID LIKE ‘123%’;

SELECT COUNT(PDID) FROM ARREST WHERE PDID LIKE ‘123%’;

SELECT COUNT(PDID) FROM PAROLE WHERE PDID LIKE ‘123%’;

SELECT COUNT(PDID) FROM PRETRIAL WHERE PDID LIKE ‘123%’;

SELECT COUNT(PDID) FROM CASEDETAIL WHERE PDID LIKE ‘123%’;

SELECT COUNT(PDID) FROM PRISONER WHERE PDID LIKE ‘123%’;

The first two queries run against the MPD database. The others run against the databases for CSOSA, PSA,DCSC and USPC respectively.

When this query is run, the application executes the first query and waits for the results to be returned before proceeding to the second query and so on to the last query. Once all the queries have been serially run, the results are returned to the user. Typical response time is about 18 seconds.

If the same query is run singly per agency (PDID begins with 123 and only the DCSC button clicked, for example) only the query that pertains to that agency (SELECT COUNT(PDID) FROM CASEDETAIL WHERE PDID LIKE ‘123%’; in this case) is executed. Typical response times are about 3 to 4 seconds.

The sum total of the response times of the 6 individual queries run singularly matches the response time of the query run across all agencies.

Note that the response times listed here take all systems components into consideration, i.e. this resultant time is the result of the operating environment: the user computer, the agency firewalls, SSL rendition, DC WAN data transfer, IIS Server response time, the database server configuration and the Inquiry application.

The JUSTIS Inquiry application clearly performs as coded within the single-threaded development environment.

The JUSTIS team realizes that users might desire lower response times. Two related points need to be addressed here:

1. A query like the one discussed here is generic and is expected to return a huge number of results. Where user queries are very specific (as would be expected in normal usage by the JUSTIS community) looking for specific cases, names, PDID's etc., response times will be lower

2. A query like the one discussed here is run across all agencies. However, it is expected that in most cases the users of the application would only be interested in data from a specific agency, not necessarily all JUSTIS agencies. Therefore, a specifically targeted query run for a specific agency or agencies will return results in less time than the generic query discussed.

The JUSTIS team also realizes that there exist times when Administrators or Super Users might need to run generic queries for administrative purposes. Lower response times than those reported here are possible in such cases but not guaranteed.

The long-term solution for such requirements is to re-code the application in a multi-threaded development environment (such as the new .NET platform, using VB.NET) to allow multiple agency queries to execute simultaneously. In that environment, the JUSTIS team would expect the results of a generic query to be returned in about 3 to 4 seconds even when run across all JUSTIS agencies. However, such an effort entails additional development, testing and deployment time and resource and infrastructure expense.

3. Summary of Analysis

Each operating component of the Inquiry Application has been analyzed and two distinct cases have emerged: one where there are problems with the CRYPTEK solution and one where the CRYPTEK solution operates as designed.

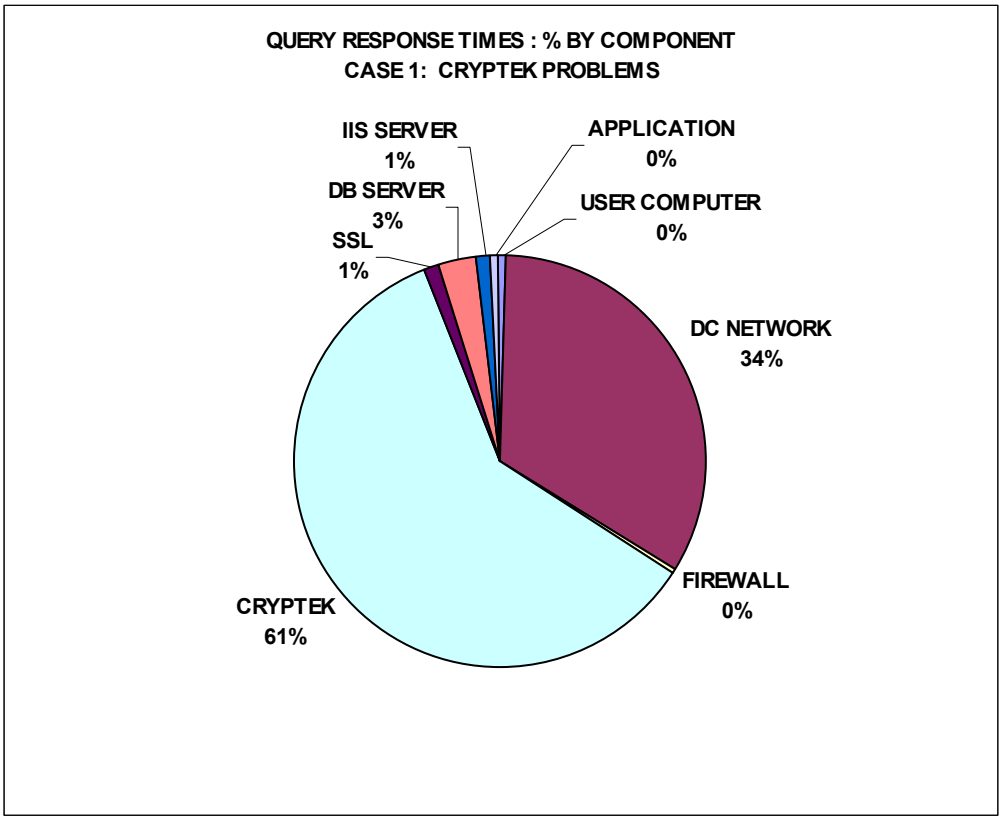
For a generic JUSTIS Inquiry (such as **PDID like '123%'**) run across ALL participating agencies, response times vary from 45+ seconds when the CRYPTEK solution fails to perform as designed to 18 seconds when the CRYPTEK solution is operating correctly.

Note that a generic query like the one described actually encompasses *six* distinct queries run against five different JUSTIS agencies.

The absolute response times (in seconds) broken up by component, for each case are as follows:

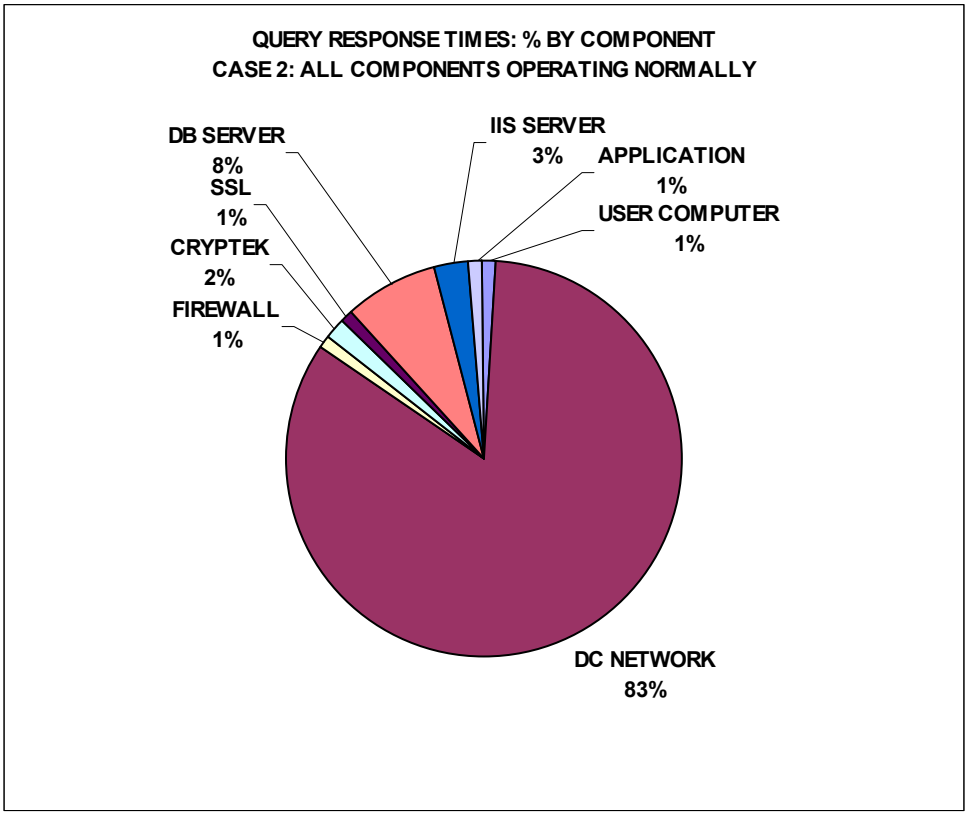
CASE 1: CRYPTEK PROBLEMS	
SYSTEM COMPONENT	**TIME ATTRITBUTABLE (SECONDS)**
USER COMPUTER	0.2
DC NETWORK	15
FIREWALL	0.2
CRYPTEK	27
SSL	0.5
DB SERVER	1.4
IIS SERVER	0.5
APPLICATION	0.2
TOTAL:	45

The Pie chart overleaf breaks up these times by the percentage of time attributable to each system component.


QUERY RESPONSE TIMES : % BY COMPONENT
CASE 1: CRYPTEK PROBLEMS
IIS SERVER
1%
APPLICATION
0%
DB SERVER
3%
USER COMPUTER
0%
SSL
1%
DC NETWORK
34%
FIREWALL
0%
CRYPTEK
61%

CASE 2: ALL SYSTEM COMPONENTS OPERATING NORMALLY	
SYSTEM COMPONENT	**TIME ATTRITBUTABLE (SECONDS)**
USER COMPUTER	0.2
DC NETWORK	15
FIREWALL	0.2
CRYPTEK	0.3
SSL	0.2
DB SERVER	1.4
IIS SERVER	0.5
APPLICATION	0.2
TOTAL:	18

The Pie chart below breaks this up by the percentage of time attributable to each system component.



The efficient working of the JUSTIS Inquiry Application requires

1. The CRYPTEK solution to work as stated
2. The DC WAN (Network) to function without any undue delays.

The JUSTIS team proposes to resolve the issues with the CRYPTEK solution as outlined in Section 2.4

The Application must be managed within the confines of the DC WAN. However, a solution to make more efficient use of the DC WAN can be implemented as outlined in Section 2.8, should this be desired.

The other operating components of the JUSTIS Inquiry Application System do not slow the application down significantly enough to warrant the need for major improvements or changes at this time.